

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

**APPENDIX III**

~~Copyright © 2000-2001 ARC International plc. All rights reserved.~~

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
#  
# Confidential Information  
# Limited Distribution to Authorized Persons Only  
# Created 2000 and Protected as an Unpublished Work  
# Under the U.S. Copyright act of 1976.  
# Copyright © 2000-2001 ARC CORES LTD  
# All Rights Reserved  
#  
# SCCS release : %M% %I% %C%  
#  
# Description : Script to analyse an ARC assembler file and  
# print frequency of usage stats for various  
# proposed ARC instruction formats  
#  
#  
#--  
===== #  
  
20 BEGIN +  
out = "c"  
#reg = "%r(0|1|2|3|13|14|15|16),"  
25 reg = "%r(0|1|2|3|13|14|15|16)([^0-9]|$)"  
regh = "%(r[0-9]+|sp|fp|gp|blink)([^0-9]|$)"  
reg01 = "%r(0|1)([^0-9]|$)"  
reg23 = "%r(2|3)([^0-9]|$)"  
reg1316 = "%r(13|14|15|16)([^0-9]|$)"  
pete = 0  
printf "" >out  
+  
  
function nxt() {  
print $0 >>out  
next  
+  
function nxte() {  
print "c" $0 >>out  
next  
+  
  
$1 == "b1" +  
b1++  
if ($2 ~ /_prolog_.*/){  
push++  
nxt()  
} else {  
calls[$2]++  
nxte()  
}  
+  
$1 == "b" +  
b++  
if ($2 ~ /_epilog_.*/){  
pop++  
nxt()  
} else {  
nxte()  
}  
+  
$1 == "beq" || $1 == "bne" +  
if ($2 ~ /_epilog_.*/){  
beq++  
nxte()  
} else {  
nxt()  
}  
+
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```

+
$1 == "bgt" || $1 == "ble" || $1 == "bge" || $1 == "bit" +
if ($2 !~ /__epilog_.*/) +
bgt++
nxtc()
+ else +
nxtc()
+
+
$1 == "bhi" || $1 == "bls" || $1 == "bhs" || $1 == "ble" +
if ($2 !~ /__epilog_.*/) +
bhi++
nxtc()
+ else +
nxtc()
+
+
$1 == "bpl" || $1 == "bmi" +
if ($2 !~ /__epilog_.*/) +
bpl++
nxtc()
+ else +
nxtc()
+
+
$1 == "jeq" || $1 == "jne" +
if ($2 ~ "blink") +
beq++
nxtc()
+
nxtc()
+
$1 == "jgt" || $1 == "jle" || $1 == "jge" || $1 == "jlt" +
if ($2 ~ "blink") +
bgt++
nxtc()
+
nxtc()
+
+
$1 == "j" +
if ($2 ~ "blink") +
jblink++
nxtc()
+
if ($2 ~ reg) +
jr++
nxtc()
+
nxtc()
+
$1 == "jl" +
if ($2 ~ reg) +
jlr++
nxtc()
+
nxtc()
+
+
$1 == "ld" +
if ($2 ~ reg) +
ld ++
if ($3 == "%fp,") +
# ldfpa[$4] ++
ldfp ++
if (($4+0) >= -32 && ($4+0) <= -4) +
ldfp32 ++
nxtc()
+
+

```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
nxt()
+
if ($3 == "%sp,") +
# ldspa[$4]++
ldsp++
nxt()
+
if ($3 == "%gp,") +
ldgp++
nxtc()
+
if ($3 ~ reg) +
# ldra[$4]++
ldr++
if ($3 ~ /\}/ || ($4 ~ /^[0-9]/ && ($4+0) >= 0 && ($4+0) < 64)) +
ldr64++
nxtc()
+
if (pete) +
if ($3 ~ /\}/ || ($3 ~ reg01 && ($4 ~ /^[0-9]/ && ($4+0) >= 0 && ($4+0) < 128))) +
ldr64p++
nxtc()
+
if ($3 ~ /\}/ || ($3 ~ reg23 && ($4 ~ /^[0-9]/ && ($4+0) >= 0 && ($4+0) < 64))) +
ldr64p++
nxtc()
+
if ($3 ~ /\}/ || ($3 ~ reg1316 && ($4 ~ /^[0-9]/ && ($4+0) >= 0 && ($4+0) < 32))) +
ldr64p++
nxtc()
+
if ($4 ~ reg) +
ldab+++
nxtc()
+
nxt()
+
if ($1 == "ldw" +
if ($2 ~ reg) +
ldw++
if ($3 == "%fp,") +
ldwfp++
if ((($4+0) >= -32 && ($4+0) <= -4) +
ldwfp32++
nxtc()
+
nxt()
+
if ($3 == "%sp,") +
ldwsp++
nxt()
+
if ($3 == "%gp,") +
ldwgp++
nxtc()
+
if ($3 ~ reg) +
ldwr++
if ($3 ~ /\}/ || ($4 ~ /^[0-9]/ && ($4+0) >= 0 && ($4+0) < 32)) +
ldwr32++
nxtc()
+
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
      if ($4 ~ reg) {
      ldwabe++
      nxt()
      +
      nxt()
      +
      $1 == "ldb" {
      if ($2 ~ reg) {
      ldb++
      if ($3 == "[%fp,") {
      ldbfp++
      if (($4+0) >= -32 && ($4+0) <= 4) {
      ldbfp32++
      nxt()
      +
      nxt()
      +
      if ($3 == "[%sp,") {
      ldbsp++
      nxt()
      +
      if ($3 == "[%gp,") {
      ldbgp++
      nxt()
      +
      if ($3 ~ reg) {
      ldbri++
      if ($3 ~ /\)/ || ($4 ~ /[^0-9]/ && ($4+0) >= 0 && ($4+0) < 16)) {
      ldbri16++
      nxtc()
      +
      if ($4 ~ reg) {
      ldbabe++
      nxt()
      +
      nxt()
      +
      +
      /st.%blink, \[%sp, 4\]/ {
      stblink++
      nxtc()
      +
      $1 == "st" {
      if ($2 ~ reg) {
      stt++
      if ($3 == "[%fp,") {
      # stfp[$4]++
      stfp++
      if (($4+0) >= -32 && ($4+0) <= 4) {
      stfp32++
      nxtc()
      +
      nxt()
      +
      if ($3 == "[%sp,") {
      # stspa[$4]++
      stsp++
      nxt()
      +
      if ($3 == "[%gp,") {
      stgp++
      nxt()
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
      +  
      + if ($3 ~ reg) +  
# stra[$4]++  
      + str++  
5   + if ($3 ~ /\$/) || ($4 ~ /[^0-9]/ && ($4+0) >= 0 && ($4+0) < 64) +  
      + str64++  
      + nxtc()  
      +  
      + nxt()  
      +  
      +  
      + $1 == "stw" +  
15  + if ($2 ~ reg) +  
      + stw++  
      + if ($3 == "%fp,") +  
# stwfpa[$4]++  
      + stwfpa++  
20  + if (($4+0) >= -32 && ($4+0) <= -4) +  
      + stwfpa32++  
      + nxt()  
      +  
      + nxt()  
      +  
      + if ($3 == "%sp,") +  
# stwspa[$4]++  
      + stwspa++  
      + nxt()  
30  +  
      + if ($3 == "%gp,") +  
      + stwgp++  
      + nxt()  
      +  
35  + if ($3 ~ reg) +  
# stwra[$4]++  
      + stwra++  
      + if ($3 ~ /\$/) || ($4 ~ /[^0-9]/ && ($4+0) >= 0 && ($4+0) < 16) +  
      + stwra16++  
40  + nxtc()  
      +  
      + nxt()  
      +  
      +  
45  + nxt()  
      +  
      + $1 == "stb" +  
      + if ($2 ~ reg) +  
      + stb++  
50  + if ($3 == "%fp,") +  
# stbfpa[$4]++  
      + stbfpa++  
      + if (($4+0) >= -32 && ($4+0) <= -4) +  
      + stbfpa32++  
55  + nxt()  
      +  
      + nxt()  
      +  
      + if ($3 == "%sp,") +  
# stbspa[$4]++  
      + stbspa++  
      + nxt()  
      +  
      + if ($3 == "%gp,") +  
# stbgp++  
      + stbgp++  
      + nxt()  
      +
```

Application No. : 09/808,469  
Filed : March 14, 2001

```
5      if ($3 ~ reg) {  
#      stbra[$4]++  
#      stbr++  
#      if ($3 ~ /\$/ || ($4 ~ /[^0-9]/ && ($4+0) >= 0 && ($4+0) < 8)) +  
#      stbr8++  
#      nxte()  
#      +  
#      nxte()  
#      +  
10     +  
#      nxte()  
#      +  
#      $1 == "mov.f" +  
#      if ($2 == "0," && $3 ~ reg) {  
#          movf0r++  
#          nxte()  
#          if ($2 == "0," && $3 ~ regh) {  
#              movf0h++  
#              nxte()  
#              +  
#              nxte()  
#              +  
#              $1 == "mov" {  
#                  if ($3 ~ /^[0-9]/) {  
#                      movi++  
#                      movia[$3]++  
#                      if ($2 ~ reg) {  
#                          if ($3 >= 0 && $3 < 64) {  
#                              movi64++  
#                              nxte()  
#                              +  
#                              if (pete) {  
#                                  if ($2 ~ reg01 && $3 >= 0 && $3 < 128) {  
#                                      movi64p++  
#                                      nxte()  
#                                      +  
#                                      if ($2 ~ reg23 && $3 >= 0 && $3 < 64) {  
#                                          movi64p++  
#                                          nxte()  
#                                          +  
#                                          if ($2 ~ reg1316 && $3 >= 0 && $3 < 32) {  
#                                              movi64p++  
#                                              nxte()  
#                                              +  
#                                              if ($3 < -256 || $3 > 255) {  
#                                                  ldrpet++  
#                                                  nxte()  
#                                                  +  
#                                                  nxte()  
#                                                  +  
#                                                  if ($3 ~ reg) {  
#                                                      if ($2 ~ reg) {  
#                                                          movr++  
#                                                          nxte()  
#                                                          +  
#                                                          if ($2 ~ reg) {  
#                                                              if ($3 ~ regh) {  
#                      movrh++  
#                      nxte()  
#                      +  
#                      if ($2 ~ regh) {  
#                          if ($3 ~ reg) {  
#                              movhr++
```

Application No. : 09/808,469  
Filed : March 14, 2001

```
    ---nxtc()
    +
    +
5   if ($3 !~ /%/ && $2 ~ reg) {
    ldrpe++
    ---nxtc()
    +
    +
10  $1 == "add" +
    if ($2 == $3 || $2 == ($3 ",") || $2 == ($4 ",")) +
    if ($4 ~ /[^0-9]/) +
    addi++
#    addia[$4]++
15  if ($3 ~ reg) +
    if ($4 >= -32 && $4 < 0) +
    subi32++
    ---nxtc()
    +
20  if ($4 >= 0 && $4 < 32) +
    addi32++
    ---nxtc()
    +
25  +
    if ($2 ~ reg && $3 ~ reg && $4 ~ reg) +
    addaab++
    ---nxtc()
    +
30  if ($2 ~ reg && $3 ~ reg && $4 ~ reg) +
    addrrh++
    ---nxtc()
    +
35  if ($2 ~ reg && $3 ~ regh && $4 ~ reg) +
    addrrh++
    ---nxtc()
    +
40  if ($4 ~ /[^0-9]/) +
    if ($2 ~ reg) +
    if ($3 ~ reg) +
    if ($4 >= -8 && $4 < 0) +
    subabi8++
    ---nxtc()
    +
45  if ($4 >= 1 && $4 <= 8) +
    addabi8++
    ---nxtc()
    +
50  +
    if ($3 ~ "%fp") +
    if ($4 >= -32 && $4 < 0) +
    addfpi32++
    ---nxtc()
    +
55  +
    if ($3 ~ /%r([12][0-9])/ && $4 >= -512 && $4 < 512) +
    addrpe++
    ---nxtc()
    +
60  +
    ---nxtc()
    +
65  if ($2 ~ reg && $3 ~ reg && $4 ~ reg) +
    addrr+++
    ---nxtc()
    +
```

**Application No.** : **09/808,469**  
**Filed** : **March 14, 2001**

```

+
$1 == "sub" +
if ($4 ~ /[^?0-9]/) +
subi++
if ($2 == $3) +
# subia[$4]++
if ($3 ~ reg) +
if ($4 >= 32 && $4 < 0) +
addi32++
nxte()
+
if ($4 >= 0 && $4 < 32) +
subi32++
nxte()
+
+
if ($2 ~ reg) +
if ($3 ~ reg) +
if ($4 >= 8 && $4 < 0) +
addabi8++
nxte()
+
if ($4 >= 1 && $4 < 8) +
subabi8++
nxte()
+
+
nxt()
+
if ($2 == $3 && $2 == ($4 ", ")) +
if ($2 ~ reg && $3 ~ reg && $4 ~ reg) +
subaaa++
nxte()
+
if ($2 ~ regh && $3 ~ regh && $4 ~ regh) +
subhhh++
nxte()
+
if ($2 ~ reg) +
subr++
if ($2 == $3) +
if ($2 ~ reg && $3 ~ reg && $4 ~ reg) +
subaab++
nxte()
+
if ($2 ~ reg && $3 ~ reg && $4 ~ regh) +
subrrh++
nxte()
+
if ($2 ~ reg && $3 ~ regh && $4 ~ reg) +
subrrrh++
nxte()
+
if ($3 ~ reg && $4 ~ reg) +
subrrrr++
nxte()
+
nxt()
+
$1 == "sub.f" +
if ($2 == "0, ") +
if ($3 ~ reg && $4 ~ reg) +

```

Application No. : 09/808,469  
Filed : March 14, 2001

```
____ cmpri++
____ nxtc()
____ }
____ if ($4 ~ /[^?0-9]/) {
____   cmpi++
____ # cmpia[$4]++
____   if ($3 ~ reg) {
____     if ($4 >= 0 && $4 < 64) {
____       cmpi64++
____       nxtc()
____     }
____     if (pete) {
____       if ($3 ~ reg01 && $4 >= 0 && $4 < 128) {
____         cmpi64p++
____         nxtc()
____       }
____       if ($3 ~ reg23 && $4 >= 0 && $4 < 64) {
____         cmpi64p++
____         nxtc()
____       }
____       if ($3 ~ reg1316 && $4 >= 0 && $4 < 32) {
____         cmpi64p++
____         nxtc()
____       }
____     }
____   }
____ }
____ }
____ +
____ nxt()
____ }
____ if ($3 ~ reg) {
____ if ($4 ~ regh) {
____   cmprh++
____   nxtc()
____ }
____ if ($3 ~ regh) {
____   if ($4 ~ reg) {
____     cmphr++
____     nxtc()
____   }
____ }
____ }
____ +
____ nxt()
____ +
____ nxt()
____ +
____ $1 == "sub.ne" {
____   if ($2 == $3 && $2 == ($4 ",,")) {
____     if ($4 ~ reg && $2 ~ reg && $3 ~ reg) {
____       subneaaaa++
____       nxtc()
____     }
____   }
____ }
____ +
____ nxt()
____ +
____ $1 == "sub.eq" {
____   if ($2 == $3 && $2 == ($4 ",,")) {
____     if ($4 ~ reg && $2 ~ reg && $3 ~ reg) {
____       subeqaaaa++
____       nxtc()
____     }
____   }
____ }
____ +
____ nxt()
____ +
____ $1 == "asl" {
____   if ($4 ~ /[^?0-9]/) {
____     asli++
____     if ($2 == $3) {
____       # aslia[$4]++
____       if ($3 ~ reg) {
____         aslia[$4]++
____       }
____     }
____   }
____ }
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
if ($4 >= 1 && $4 <= 8) {  
    asli8++  
}  
+  
if ($4 >= 1 && $4 < 32) {  
    asli32++  
}  
+  
nxtc()  
+  
if ($2 ~ reg) {  
    if ($3 ~ reg && $4 >= 2 && $4 < 3) {  
        aslab2++  
        nxtc()  
    }  
}  
+  
nxt()  
+  
if ($4 ~ reg && $2 ~ reg && $3 ~ reg) {  
    aslaab++  
    nxtc()  
}  
+  
if ($2 ~ reg && $3 ~ reg && $4 != reg) {  
    aslab1++  
    nxtc()  
}  
+  
$1 == "asr" {  
    if ($4 ~ /[^?0-9]/) {  
        asri++  
        if ($2 == $3) {  
            # asria[$4]++  
            if ($3 ~ reg) {  
                if ($4 >= 1 && $4 <= 8) {  
                    asri8++  
                }  
                if ($4 >= 1 && $4 < 32) {  
                    asri32++  
                }  
            }  
            nxtc()  
        }  
    }  
    if ($2 ~ reg) {  
        if ($3 ~ reg && $4 >= 2 && $4 < 3) {  
            asrab2++  
            nxtc()  
        }  
    }  
    nxt()  
}  
+  
if ($4 ~ reg && $2 ~ reg && $3 ~ reg) {  
    asraab++  
    nxtc()  
}  
+  
if ($2 ~ reg && $3 ~ reg && $4 != reg) {  
    asrab1++  
    nxtc()  
}  
+  
$1 == "lcr" {  
    if ($4 ~ /[^?0-9]/) {  
        lcri++  
        if ($2 == $3) {  
            # lcria[$4]++  
            if ($3 ~ reg) {  
                if ($4 >= 1 && $4 <= 8) {  
                    lcri8++  
                }  
            }  
        }  
    }  
}
```

Application No. : 09/808,469  
Filed : March 14, 2001

```
      if ($4 >= 1 && $4 < 32) +  
      lsr132++  
      +  
      nxtc()  
5      +  
      +  
      if ($2 ~ reg) +  
      if ($3 ~ reg && $4 >= 2 && $4 < 3) +  
      lsrab2++  
10     nxtc()  
      +  
      +  
      +  
      nxt()  
15     +  
      if ($4 ~ reg && $2 ~ reg && $3 ~ reg) +  
      lsrab1++  
      nxtc()  
      +  
      +  
      if ($2 ~ reg && $3 ~ reg && $4 !~ reg) +  
      lsrab1++  
      nxtc()  
      +  
      +  
25     $1 == "mul64" +  
      if ($2 == "0,") +  
      if ($4 ~ /[^?0-9]/) +  
      muli++  
      # mulia[$4]++  
      if ($3 ~ reg) +  
      if ($4 >= 0 && $4 < 32) +  
      muli32++  
      nxtc()  
      +  
      +  
35     if ($3 ~ reg && $4 ~ reg) +  
      mul0ab++  
      nxtc()  
      +  
      +  
40     nxt()  
      +  
      $1 == "and.f" +  
      if ($2 == "0,") +  
45     if ($4 ~ /[^?0-9]/) +  
      andfi++  
      # andfia[$4]++  
      if ($3 ~ reg) +  
      if ($4 >= 0 && $4 < 32) +  
50     andfi32++  
      nxtc()  
      +  
      +  
55     if ($3 ~ reg && $4 ~ reg) +  
      andfab++  
      nxtc()  
      +  
      +  
60     nxt()  
      +  
      $1 == "and" +  
      if ($2 == $3 || $2 == ($3 ",") || $2 == ($4 ",")) +  
65     if ($4 ~ /[^?0-9]/) +  
      andi++  
      # andia[$4]++  
      if ($3 ~ reg) +
```

Application No. : 09/808,469  
Filed : March 14, 2001

```
      if ($4 >= 0 && $4 < 32) {  
      andi32++  
      nxte()  
      }  
      +  
      +  
      if ($2 ~ reg && $3 ~ reg && $4 ~ reg) {  
      andaabb++  
      nxte()  
      }  
      +  
      if ($2 ~ reg && $3 ~ reg && $4 ~ reg) {  
      andrrrr++  
      nxte()  
      }  
      +  
      +  
      $1 == "extb" {  
      if ($2 == ($3 ",,")) {  
      if ($2 ~ reg && $3 ~ reg) {  
      extbr++  
      nxte()  
      }  
      +  
      +  
      nxte()  
      }  
      +  
      $1 == "extw" {  
      if ($2 == ($3 ",,")) {  
      if ($2 ~ reg && $3 ~ reg) {  
      extwr++  
      nxte()  
      }  
      +  
      +  
      nxte()  
      }  
      +  
      $1 == "sexb" {  
      if ($2 == ($3 ",,")) {  
      if ($2 ~ reg && $3 ~ reg) {  
      sexbr++  
      nxte()  
      }  
      +  
      +  
      nxte()  
      }  
      +  
      $1 == "sexw" {  
      if ($2 == ($3 ",,")) {  
      if ($2 ~ reg && $3 ~ reg) {  
      sexwr++  
      nxte()  
      }  
      +  
      +  
      nxte()  
      }  
      +  
      +  
      ($2 == $3 || $2 == ($3 ",,") || $2 == ($4 ",,")) {  
      if ($1 == "add" || $1 == "sub" || $1 == "and" || $1 == "or" || $1 == "xor" || $1 ==  
      "asl" || $1 == "asr" || $1 == "lsl") {  
      if ($2 ~ reg) {  
      if ($2 == $3) {  
      if ($4 ~ reg) {  
      opaab[$1]++  
      nxte()  
      }  
      +  
      } else {  
      if ($3 ~ reg && $2 == ($4 ",,")) {  
      opaab[$1]++  
      nxte()  
      }  
      +  
      }
```

Application No. : 09/808,469  
Filed : March 14, 2001

```
+  
+  
+  
5 +  
+  
+  
nxt()  
# print $0  
+  
10 END {  
if (1)  
OPS = "\t"  
# print "\nopaab"  
for (i in opaab){  
if (i == "add" || i == "sub" || i == "and" || i == "or" || i == "xor" || i == "asl"  
|| i == "asr" || i == "lsr") {  
print i, opaab[i], int(opiaab[i]*1000/NR)/10  
+  
+  
+  
20 # print "\nldfpa"  
# for (i in ldfpa) print i, ldfpa[i]  
# print "\nstfpa"  
# for (i in stfpa) print i, stfpa[i]  
# print "\nldr0a"  
25 # for (i in ldr0a) print i, ldr0a[i]  
# print "\nmovia"  
# for (i in movia) print i, movia[i]  
# print "\nnaddia"  
# for (i in addia) print i, addia[i]  
30 # print "\nsubia"  
# for (i in subia) print i, subia[i]  
# print "\ncmpia"  
# for (i in cmpia) print i, cmpia[i]  
+  
35 for (i in calls) {  
# print i, calls[i]  
if (calls[i] > 1) {  
calls2 += (calls[i]-2)  
+  
40 callsall += calls[i]  
+  
# print "callsall", callsall, int(callsall*1000/NR)/10  
# print "calls2", calls2, int(calls2*1000/NR)/10  
+  
45 # bl = calls2  
bl = bl - push  
b = b - pop  
print "bl", bl, int(bl*1000/NR)/10  
# print "push", push, int(push*1000/NR)/10  
50  
print "b", b, int(b*1000/NR)/10  
# print "pop", pop, int(pop*1000/NR)/10  
print "beq", beq, int(beq*1000/NR)/10  
print "bgt", bgt, int(bgt*1000/NR)/10  
55 print "bhi", bhi, int(bhi*1000/NR)/10  
print "bpl", bpl, int(bpl*1000/NR)/10  
+  
print "stblink", stblink, int(stblink*1000/NR)/10  
print "jblink", jblink, int(jblink*1000/NR)/10  
60 print "jr", jr, int(jr*1000/NR)/10  
print "jlr", jlr, int(jlr*1000/NR)/10  
+  
print "movr", movr, int(movr*1000/NR)/10  
print "movf0r", movf0r, int(movf0r*1000/NR)/10  
65 print "movf0h", movf0h, int(movf0h*1000/NR)/10  
print "movrh", movrh, int(movrh*1000/NR)/10  
print "movhr", movhr, int(movhr*1000/NR)/10
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
5      print "cmprh", cmprh, int(cmprh*1000/NR)/10
      print "cmphr", cmphr, int(cmphr*1000/NR)/10
      print "cmpr", cmpr, int(cmpr*1000/NR)/10
10     print "cmpl64", cmpl64, int(cmpl64*1000/NR)/10
      print "cmpl64p", cmpl64p, int(cmpl64p*1000/NR)/10
      print "movi64", movi64, int(movi64*1000/NR)/10
      print "movi64p", movi64p, int(movi64p*1000/NR)/10
15     print "addi32", addi32, int(addi32*1000/NR)/10
      print "subi32", subi32, int(subi32*1000/NR)/10
20     print "addabi8", addabi8, int(addabi8*1000/NR)/10
      print "subabi8", subabi8, int(subabi8*1000/NR)/10
      print "subneaaa", subneaaa, int(subneaaa*1000/NR)/10
      print "subeqaaa", subeqaaa, int(subeqaaa*1000/NR)/10
25     print "subhhh", subhhh, int(subhhh*1000/NR)/10
      print "subaaa", subaaa, int(subaaa*1000/NR)/10
      print "subaab", subaab, int(subaab*1000/NR)/10
      print "subrrr", subrrr, int(subrrr*1000/NR)/10
      print "addaab", addaab, int(addaab *1000/NR)/10
      print "addr", addr, int(addr *1000/NR)/10
25     print "addrh", addrh, int(addrh *1000/NR)/10
      print "asli8", asli8, int(asli8*1000/NR)/10
#     print "asli32", asli32, int(asli32*1000/NR)/10
30     print "aslalb1", aslalb1, int(aslalb1*1000/NR)/10
      print "aslalb2", aslalb2, int(aslalb2*1000/NR)/10
      print "aslaab", aslaab, int(aslaab*1000/NR)/10
35     print "asri8", asri8, int(asri8*1000/NR)/10
#     print "asri32", asri32, int(asri32*1000/NR)/10
      print "asrab1", asrab1, int(asrab1*1000/NR)/10
      print "asrab2", asrab2, int(asrab2*1000/NR)/10
      print "asraab", asraab, int(asraab*1000/NR)/10
40     print "lsri8", lsri8, int(lsri8*1000/NR)/10
#     print "lsri32", lsri32, int(lsri32*1000/NR)/10
      print "lsrab1", lsrab1, int(lsrab1*1000/NR)/10
      print "lsrab2", lsrab2, int(lsrab2*1000/NR)/10
      print "lsraab", lsraab, int(lsraab*1000/NR)/10
45     print "andi32", andi32, int(andi32*1000/NR)/10
      print "andfi32", andfi32, int(andfi32*1000/NR)/10
      print "andaab", andaab, int(andaab *1000/NR)/10
      print "andfab", andfab, int(andfab *1000/NR)/10
50     print "mul0ab", mul0ab, int(mul0ab *1000/NR)/10
      print "muli32", muli32, int(muli32 *1000/NR)/10
      print "ldabe", ldabe, int(ldabe *1000/NR)/10
      print "ldbabe", ldbabe, int(ldbabe *1000/NR)/10
55     print "ldwabe", ldwabe, int(ldwabe *1000/NR)/10
      print "ldr64", ldr64, int(ldr64 *1000/NR)/10
      print "ldr64p", ldr64p, int(ldr64p *1000/NR)/10
      print "ldwr32", ldwr32, int(ldwr32 *1000/NR)/10
60     print "ldbri16", ldbri16, int(ldbri16 *1000/NR)/10
      print "str64", str64, int(str64 *1000/NR)/10
      print "stbr8", stbr8, int(stbr8 *1000/NR)/10
      print "stwr16", stwr16, int(stwr16 *1000/NR)/10
65     print "ldrpe", ldrpe, int(ldrpe *1000/NR)/10
      print "addrpe", addrpe, int(addrpe *1000/NR)/10
```

**Application No. : 09/808,469**  
**Filed : March 14, 2001**

```
-print "ldfp32", ldfp32, int(ldfp32*1000/NR)/10
#print "stfp32", stfp32, int(stfp32*1000/NR)/10
#print "addfpi32", addfpi32, int(addfpi32*1000/NR)/10

5   -print "ldgp", ldgp, int(ldgp*1000/NR)/10
#print "stgp", stgp, int(stgp*1000/NR)/10

-print "extbr", extbr, int(extbr*1000/NR)/10
10  -print "extwr", extwr, int(extwr*1000/NR)/10
#print "sexbr", sexbr, int(sexbr*1000/NR)/10
#print "sexwr", sexwr, int(sexwr*1000/NR)/10

15  # print "movi", movi, "movi64", movi64, "movi128", movi128
# print "addi", addi, "addi32", addi32, "addi64", "addi128", addi128
# print "subi", subi, "subi32", subi32, "subi64", subi64, "subi128"
+
+
#function p(a, b) {
# print "a", b, int(b*100/NR)
20  #
#}

25  #/(j|jl|bl)(ge|gt|le|lt|ne|eq|pl|mi|hi|hsl|lo|ls)?\.\d/ {
# stored = $0
# sub(/\.d/, "", stored)
# getline
# print $0
# print stored
# nxte()
#
30  #
#{ print $0 }
```